



Zadání bakalářské práce

Název:	Automatizace administrátorských procesů databáze aplikace Anketa ČVUT
Student:	Alice Kopalová
Vedoucí:	Ing. Michal Valenta, Ph.D.
Studijní program:	Informatika
Obor / specializace:	Webové a softwarové inženýrství, zaměření Softwarové inženýrství
Katedra:	Katedra softwarového inženýrství
Platnost zadání:	do konce letního semestru 2023/2024

Pokyny pro vypracování

V současné verzi aplikace Anketa ČVUT je řada činností administrátora, např. vytvoření nového běhu semestru či vyhodnocení uzavřených anket, realizována formou spouštění SQL skriptů. Cílem práce je navrhnout a implementovat postup, který umožní provádět tyto činnosti ze správcovské aplikace.

Postupujte v těchto krocích:

- zdokumentujte stávající stav činností administrátora, které budou předmětem automatizace,
- diskutujte možnosti, jak tyto činnosti spustit z kódu backend části správcovské aplikace Anketa ČVUT, jednu metodu zvolte,
- navrhnete a implementujte automatizaci zvolených činností administrátora; minimálně pro tyto činnosti: příprava nového běhu, tedy obnova materializovaných pohledů a kopie anket, vyhodnocení anket; případně přidejte refaktoring materializovaných pohledů a plánování jejich automatizované obnovy,
- postupy zdokumentujte.



**FAKULTA
INFORMAČNÍCH
TECHNOLÓGIÍ
ČVUT V PRAZE**

Bakalářská práce

Automatizace administrátorských procesů databáze aplikace Anketa ČVUT

Alice Kopalová

Katedra softwarového inženýrství

Vedoucí práce: Ing. Michal Valenta, Ph.D.

11. května 2023

Poděkování

Ráda bych poděkovala vedoucímu práce Ing. Michalovi Valentovi, Ph.D., za jeho vedení, podporu a cenné rady, které mi pomohly při psaní této práce. Dále bych chtěla poděkovat své rodině za jejich lásku a podporu během psaní této práce i během studia.

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona, ve znění pozdějších předpisů. V souladu s ust. § 2373 odst. 2 zákona č. 89/2012 Sb., občanský zákoník, ve znění pozdějších předpisů, tímto uděluji nevýhradní oprávnění (licenci) k užití této mojí práce, a to včetně všech počítačových programů, jež jsou její součástí či přílohou a veškeré jejich dokumentace (dále souhrnně jen „Dílo“), a to všem osobám, které si přejí Dílo užít. Tyto osoby jsou oprávněny Dílo užít jakýmkoli způsobem, který nesnižuje hodnotu Díla a za jakýmkoli účelem (včetně užití k výdělečným účelům). Toto oprávnění je časově, teritoriálně i množstevně neomezené. Každá osoba, která využije výše uvedenou licenci, se však zavazuje udělit ke každému dílu, které vznikne (buť jen zčásti) na základě Díla, úpravou Díla, spojením Díla s jiným dílem, zařazením Díla do díla souborného či zpracováním Díla (včetně překladu) licenci alespoň ve výše uvedeném rozsahu a zároveň zpřístupnit zdrojový kód takového díla alespoň srovnatelným způsobem a ve srovnatelném rozsahu, jako je zpřístupněn zdrojový kód Díla.

V Praze dne 11. května 2023

.....

České vysoké učení technické v Praze
Fakulta informačních technologií

© 2023 Alice Kopalová. Všechna práva vyhrazena.

Tato práce vznikla jako školní dílo na Českém vysokém učení technickém v Praze, Fakultě informačních technologií. Práce je chráněna právními předpisy a mezinárodními úmluvami o právu autorském a právech souvisejících s právem autorským. K jejímu užití, s výjimkou bezúplatných zákonných licencí a nad rámec oprávnění uvedených v Prohlášení na předchozí straně, je nezbytný souhlas autora.

Odkaz na tuto práci

Kopalová, Alice. *Automatizace administrátorských procesů databáze aplikace Anketa ČVUT*. Bakalářská práce. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2023.

Abstrakt

Anketa ČVUT je webová aplikace, pomocí které České vysoké učení technické v Praze získává zpětnou vazbu od studentů. V současné době aplikaci chybí nástroj pro spouštění administrátorských činností, jako je aktualizace dat v databázi před spuštěním ankety a vyhodnocení ankety po jejím ukončení. Tyto procesy jsou realizovány spuštěním skriptů, které je nutné ručně upravit pro každý běh ankety a spustit přímo v databázi aplikace.

Náplní této práce je analýza a dokumentace administrátorských procesů v aplikaci Anketa ČVUT, návrh řešení, jak administrátorské procesy spouštět z backendu aplikace, a jeho implementace.

Výstupem práce je rozšíření Oracle databáze, kterou aplikace používá, o spustitelné procedury, které administrátorské procesy zprostředkovávají. Aplikaci tak bude v budoucnu možné rozšířit o nástroj s uživatelským rozhraním, ze kterého bude možné administrátorské procesy spouštět. Dalším přínosem této práce je pak dokumentace některých dosud nepopsaných schémat v databázi aplikace Anketa ČVUT.

Klíčová slova Anketa ČVUT, databáze, PL/SQL, administrace, automatizace, Oracle

Abstract

Survey CTU is a web application with which the Czech Technical University in Prague receives feedback from students. Currently, the application lacks a tool for running administrative activities, such as updating data in the database before starting the poll and evaluating the poll after it ends. These processes are implemented by running scripts that must be manually modified for each survey run and run directly in the application database.

The content of this work is the analysis and documentation of administrative processes in the CTU Survey application, the proposal of a solution for how to run administrative processes from the backend of the application, and its implementation.

The output of the work is the extension of the Oracle database used by the application with executable procedures that are mediated by administrative processes. In the future, it will be possible to extend the application's administrative module with a tool from which administrative processes can be started. Another benefit of this work is the documentation of some previously undescribed schemas in the database of the CTU Survey application.

Keywords Survey CTU, database, PL/SQL, administration, automation, Oracle

Obsah

Úvod	1
1 Business procesy v databázi	3
1.1 SQL	3
1.2 PL/SQL	4
1.3 Dynamic SQL	4
1.4 Oracle Scheduler	5
2 Aplikace Anketa ČVUT	7
2.1 Popis aplikace	7
2.1.1 Pojmy	7
2.1.2 Průběh ankety	8
2.1.3 Popis anketního lístku	9
2.1.4 Architektura aplikace	10
3 Administrátorské procesy v databázi	11
3.1 Popis databáze	11
3.1.1 Databázový model ankety	11
3.2 Administrátorské činnosti v databázi	14
3.2.1 Vyhodnocení ankety	14
3.2.2 Příprava ankety	18
3.2.3 Současné řešení	19
3.2.4 Problémy se současným řešením	19
4 Návrh řešení	21
4.1 Požadavky	22
4.2 Výběr technologií	23
5 Implementace	25
5.1 Vývojové nástroje	25

5.1.1	Oracle SQL Developer	25
5.1.2	Vývojové prostředí	25
5.2	Implementace procedur	25
5.2.1	Implementace DDL příkazů	26
5.2.2	Parametrizace názvů tabulek	27
5.2.3	Ošetření parametrů	29
5.2.4	Logovací procedura	31
5.3	Testování a nasazení	31
6	Závěr	33
	Bibliografie	35

Seznam obrázků

2.1	Anketní lístek předmětové ankety.	9
2.2	Anketní lístek nepředmětové ankety.	10
3.1	ER model databázového schématu ankety.	12
3.2	Tabulky pro vyhodnocení předmětové ankety.	15
3.3	Tabulky pro vyhodnocení nepředmětové ankety.	16
3.4	Tabulky pro archivaci odpovědí z anketních lístků.	17
3.5	Tabulky pro archivaci vyplněnosti předchozích anket.	18
4.1	Náčrt uživatelského rozhraní pro vyhodnocení ankety.	21
5.1	Skripty pro vyhodnocení předmětové ankety na jednotlivých fakultách pro letní semestr ročníku 2021/2022.	28
5.2	SQL Injection varování v editoru Oracle SQL Developer v proceduře course_eval_create.	29
5.3	Tabulky obsahující klíče předmětových a učitelských otázek pro potřeby procedury course_eval_create.	31
5.4	Tabulky pro záznamy proběhlých procedur.	31

Seznam tabulek

5.1	Struktura balíčků s novými procedurami.	26
-----	---	----

Úvod

Anketa ČVUT je webová aplikace, pomocí které České vysoké učení technické v Praze získává zpětnou vazbu od studentů k předmětům, vyučujícím a fungování svých fakult. Výsledky hodnocení pak slouží jako zpětná vazba vyučujícím k náplni a způsobu výuky jejich předmětů, jako informace pro vedení fakulty o fungování fakulty a výkonu jejich zaměstnanců a jako doporučení pro ostatní studenty, jaké předměty si zapsat a co od zapsaných předmětů mohou čekat. Anketa probíhá vždy na konci semestru, a pokud student neuvede jinak, pak jsou všechny odpovědi v anketě anonymní.

V současné době aplikaci chybí nástroj pro spouštění administrátorských činností, jako je aktualizace dat v databázi před spuštěním ankety a vyhodnocení ankety po jejím ukončení. Tyto procesy jsou realizovány spuštěním skriptů, které je nutné ručně upravit pro každý běh ankety a spustit přímo v databázi aplikace.

Výstupem této práce bude začlenění funkcionalit skriptů přímo do aplikace Anketa ČVUT tak, aby bylo možné administrátorské procesy spouštět z backendu aplikace. To následně umožní implementaci nástroje, ze kterého bude možné připravovat a vyhodnocovat anketu vzdáleně a bez nutnosti modifikace zdrojového kódu.

Konkrétní cíle této práce jsou následující:

1. Analyzovat a zdokumentovat administrátorské procesy probíhající v aplikaci Anketa ČVUT.
2. Navrhnout a diskutovat možnosti, jak spouštět administrátorské procesy z backendu aplikace.
3. Jednu z možností zvolit, implementovat a zdokumentovat.

Business procesy v databázi

Následující kapitola popisuje způsoby, jak lze zpracovávat data v relačních databázích. Pro potřeby této práce se jedná konkrétně o databáze postavené na technologii Oracle.

1.1 SQL

SQL je dotazovací jazyk, který slouží k přístupu k datům v databázi, k jejich vytváření, modifikaci a mazání a ke správě databáze. Jazyk SQL lze rozdělit do několika částí.[1]

Data Definition Language

DDL příkazy jazyka SQL umožňují definici nových tabulek, pohledů a materializovaných pohledů a jejich integritních omezení, jejich modifikaci a mazání. Také umožňují upravovat přístupová práva k objektům v databázi. Tato skupina zahrnuje příkazy jako CREATE, DROP, ALTER, GRANT a REVOKE.

Data Manipulation Language

DML příkazy umožňují formulovat dotazy nad již existujícími tabulkami a pohledy včetně modifikace a odstranění existujících záznamů nebo vložení nových. Tato skupina SQL příkazů zahrnuje příkazy jako SELECT, INSERT, MERGE, UPDATE a DELETE.

Transaction Control Language

Tato skupina příkazů umožňuje spravovat změny v databázi způsobené DML příkazy. Do této skupiny patří příkazy COMMIT a ROLLBACK.

1.2 PL/SQL

PL/SQL je procedurální rozšíření jazyka SQL pro Oracle databáze. Kromě SQL příkazů tak umožňuje například deklaraci proměnných, iteraci a řízení běhu programu pomocí podmínek.

Program v jazyce PL/SQL je strukturován do bloku, který je rozdělený do několika sekcí.

Deklarační sekce Program v jazyce PL/SQL začíná deklarační sekcí, což je jediná část programu, kde je možné deklarovat proměnné, definovat nové datové typy a alokovat paměť.

Výkonná sekce Výkonná sekce PL/SQL programu je uvozena klíčovým slovem BEGIN a obsahuje samotný spustitelný kód. Tato sekce je v PL/SQL bloku povinná a musí vždy obsahovat alespoň jeden příkaz.

Sekce pro zpracování výjimek Tato sekce umožňuje zpracovat výjimky, které se vyskytly ve výkonné sekci programu.

```
DECLARE
    numerator NUMBER := 5;
    denominator NUMBER := 0;
BEGIN
    result := numerator / denominator;
EXCEPTION
    WHEN ZERO_DIVIDE THEN
        DBMS_OUTPUT.PUT_LINE( 'Division by zero' );
END;
```

Listing 1: PL/SQL blok s deklarací proměnných, výkonným kódem a zpracováním výjimky.

Tímto způsobem lze vytvářet anonymní PL/SQL skripty, které je pak spustitelné v Oracle databázi. Anonymní skripty je možné pojmenovat a uložit jako databázové objekty, které pak lze volat v databázi pomocí jejich jména. Tyto programové jednotky uložené v databázi se nazývají procedury či funkce.¹[2]

Přestože jazyk PL/SQL je rozšířením jazyka SQL, kvůli způsobu, jakým je kompilován, není v PL/SQL blocích možné provádět DDL příkazy bez použití Dynamic SQL.[3]

1.3 Dynamic SQL

Dynamic SQL je rozšířením jazyka SQL, které umožňuje konstruovat a spouštět SQL příkazy až za běhu programu, lze ho tak využít pro provádění DDL

¹Funkce se od procedur liší tím, že vrací hodnotu.

```
CREATE OR REPLACE PROCEDURE divide_numbers(  
    numerator IN NUMBER, denominator IN NUMBER  
)  
AS  
BEGIN  
    result := numerator / denominator;  
EXCEPTION  
    WHEN ZERO_DIVIDE THEN  
        DBMS_OUTPUT.PUT_LINE( 'Division by zero' );  
END;
```

Listing 2: PL/SQL blok zapsaný jako procedura v databázi jménem divide_numbers.

příkazů v PL/SQL blocích a konstrukci SQL příkazů, jejichž tvar není plně znám při spuštění programu.

Dynamic SQL lze provádět pomocí nativní verze Dynamic SQL v podobě příkazu EXECUTE IMMEDIATE nebo funkcí obsažených v balíčku DBMS_SQL.[4]

1.4 Oracle Scheduler

Oracle Scheduler je nástroj pro plánování a automatizaci předdefinovaných činností implementovaný pomocí balíčku DBMS_SCHEDULER. Umožňuje spouštět uložené procedury, anonymní PL/SQL bloky a externí spustitelné soubory jak na lokálním, tak na vzdáleném zařízení a zaznamenává jejich průběh a výsledek. Scheduler rovněž umožňuje rozdělovat činnosti do skupin s různou prioritou, která je pak reflektována při rozdělování prostředků za běhu. Plánované činnosti mohou být jednorázové, nebo se mohou opakovat. Spuštění činnosti lze plánovat podle:

- data a času, například každou středu v pět hodin ráno,
- události, například při selhání transakce,
- výsledku předcházející činnosti, pokud jsou součástí předem definované posloupnosti činností.

Tento nástroj tak umožňuje úplnou automatizaci procesů v databázi.

Aplikace Anketa ČVUT

Následující kapitola se věnuje popisu fungování aplikace Anketa ČVUT.

2.1 Popis aplikace

Anketa ČVUT je webová aplikace, pomocí které ČVUT získává zpětnou vazbu od studentů.

Na základě studentských hodnocení vyučující zjišťují, co si o předmětech a jejich způsobu výuky myslí studenti, vedení fakulty získává informace o tom, jak jsou studenti spokojeni s jejím fungováním a výkonem jejích zaměstnanců, a sami studenti se mohou podle ankety rozhodnout, jaké předměty si chtějí zapsat a co mohou od zapsaných předmětů a jejich vyučujících očekávat.[5]

Hodnocení probíhá na konci semestru. Studenti se v anketě mohou vyjádřit k předmětům, které si zapsali, k jejich vyučujícím a k fungování fakulty jako celku. Anketa umožňuje studentům vyjádřit se slovně nebo použít známkování pomocí emotikonů na škále od 1 do 5. Pokud student neuvede jinak, pak jsou všechny odpovědi v anketě anonymní.

2.1.1 Pojmy

V následující části jsou definovány pojmy, které budou používány k popisu aplikace Anketa ČVUT a jejího fungování.[6]

Anketa Anketa je způsob realizace vnitřního hodnocení na ČVUT. V rámci každé fakulty probíhá na konci každého semestru souběžně předmětová a nepředmětová anketa.

Předmětová anketa V předmětové anketě studenti hodnotí předměty zapsané v daném semestru. Pro každý předmět student vyplňuje jeden anketní lístek, kde může zhodnotit celkový dojem z předmětu, jeho náplň a zakončení a ohodnotit jednotlivé vyučující.

2. APLIKACE ANKETA ČVUT

Nepředmětová anketa V nepředmětové anketě se studenti vyjadřují k fungování fakulty jako celku. Student vyplňuje jeden anketní lístek pro každou fakultu, na které měl v semestru zapsaný alespoň jeden předmět.

Anketní lístek Anketní lístek je krátký dotazník obsahující množinu hodnotových a textových otázek.

Otázka Otázky mohou být buď textové, hodnotové, nebo kombinované. Odpovědí na hodnotovou otázku je známka na škále od 1 do 5 reprezentovaná pomocí emotikonů. Odpovědí na textovou otázku je prostý text. Kombinovaná otázka umožňuje oba druhy odpovědí.

Správce systému Správce systému je zodpovědný za provoz aplikace včetně přípravy, spuštění a vyhodnocení anket. V současné době je jediným správcem systému Ing. Michal Valenta, Ph.D.

Fakultní správce Fakultní správce je zodpovědný za běh ankety na příslušné fakultě. Má možnost volit dodatečná nastavení ankety, jako například vyloučení některých předmětů nebo učitelů z hodnocení.

KOS Studijní informační systém, který obsahuje data o studentech, jejich předmětech a výsledcích.

2.1.2 Průběh ankety

Anketa během svého běhu prochází několika fázemi, které udávají, jaké akce v anketě právě probíhají a kdo k ní má přístup.

Fáze P Ve fázi přípravy má k anketě přístup pouze správce systému a fakultní správci. Správce systému provede přípravu a aktualizaci dat pro vytvoření nového běhu ankety. Fakultní správci pak mají možnost dodatečných úprav pro potřeby svojí fakulty.

Fáze U Ve fázi vyplňování v módu U mohou studenti v předmětové anketě vyplňovat pouze anketní lístky předmětů, které již úspěšně dokončili. Nepředmětová anketa je v této fázi zpřístupněna vždy.

Fáze Z Ve fázi vyplňování v módu Z mohou studenti v předmětové anketě vyplňovat anketní lístky všech předmětů, které měli v daném semestru zapsané bez ohledu na způsob zakončení předmětu. Nepředmětová anketa je v této fázi zpřístupněna vždy.

Fáze K Ve fázi uzavření ankety již není možné vyplňovat anketní lístky. Dochází k vyhodnocení ankety a skrytí nevhodných studentských komentářů. Vyučující mají možnost vkládat reakce ke studentským komentářům.

2.1.3 Popis anketního lístku

Sestava otázek v anketním lístku se liší podle toho, zda je anketa předmětová, nebo nepředmětová, a na jaké fakultě probíhá. [7, 8] Typ, znění a pořadí otázek v předmětové nebo nepředmětové anketě udává fakultní správce, otázky v anketním lístku je tak možné přizpůsobit potřebám fakulty.

K odpovědím na otázky může student připojit i některé ze svých identifikačních údajů. Student může vybrat libovolný počet z následujících údajů:

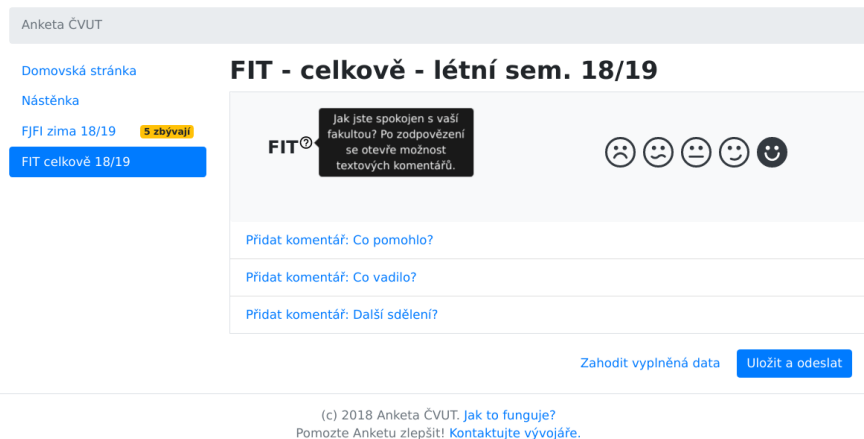
- Mé jméno
- Můj obor studia
- Můj ročník studia
- Můj studijní průměr
- Role předmětu²

The screenshot shows a web interface for evaluating a subject. The main heading is 'Hodnotíte 04AM1 Angličtina M1'. On the left, there is a sidebar with navigation links: 'Anketa ČVUT', 'Domovská stránka', 'Nástěnka', 'FJFI zima 18/19' (with a 'Zobnovit' button), and 'FIT celkově 18/19'. The main content area contains a 'Celkový dojem' (Overall impression) section with a dropdown menu for 'Ohodnotte celkový dojem z předmětu (obsah+vyučující). Níže pak můžete přidat hodnocení předmětu a/nebo vyučujících'. Below this are three rows for individual evaluators: 'Čápová Hana Mgr.', 'Fučík Radek Ing. Ph.D.', and 'Máca Radek Ing. Ph.D.', each with a rating scale of five smiley faces. At the bottom, there are links to add comments for each category: 'Obsah předmětu', 'Zakončení předmětu', 'Čápová Hana Mgr.', 'Fučík Radek Ing. Ph.D.', and 'Máca Radek Ing. Ph.D.'. A 'Přidat hodnocení vyučujícího' button is also present. At the very bottom, there are buttons for 'Zahodit vyplněná data' and 'Uložit a odeslat'.

Obrázek 2.1: Anketní lístek předmětové ankety.

²Zda je tento předmět povinný ve studijním programu studenta.

2. APLIKACE ANKETA ČVUT



Obrázek 2.2: Anketní lístek nepředmětové ankety.

2.1.4 Architektura aplikace

Současná podoba aplikace byla navržena a implementována v rámci závěrečné práce Vojtěcha Štěchy „Anketa ČVUT - verze 3.0 - vyplňování anketních lístků“ [9].

Architektura aplikace Anketa ČVUT je postavená na principu třívrstvé architektury [10] a sestává ze tří komponent:

- uživatelského rozhraní, které umožňuje vyplňování anketních lístků v době běhu ankety a zobrazení výsledků ankety po jejím vyhodnocení, implementovaného v JavaScriptovém frameworku React,
- serverové části aplikace pro zpracování dat implementované v jazyce Java za použití frameworku Spring,
- databáze využívající technologii Oracle, která slouží pro uchovávání informací o anketách, dat z anketních lístků a vyhodnocení ukončených anket.

V rámci závěrečné práce Jakuba Juna „Anketa ČVUT verze 3 - modul pro správu anket - uživatelské rozhraní“ [11] byla aplikace rozšířena o nástroj, který umožňuje správu anket, jako například tvorbu, modifikaci a smazání otázky či anketního lístku, nastavení předmětů, které jsou vyloučeny z předmětové ankety apod.

Administrátorské procesy v databázi

V současné době administrátorské ankety probíhají v databázi aplikace. Tato kapitola se nejprve věnuje analýze databázového schématu ankety a poté popisuje činnosti spojené s přípravou ankety před spuštěním a vyhodnocením ankety po jejím ukončení. V závěru kapitoly je popsána implementace těchto činností a důvody, proč je současné řešení nevyhovující.

3.1 Popis databáze

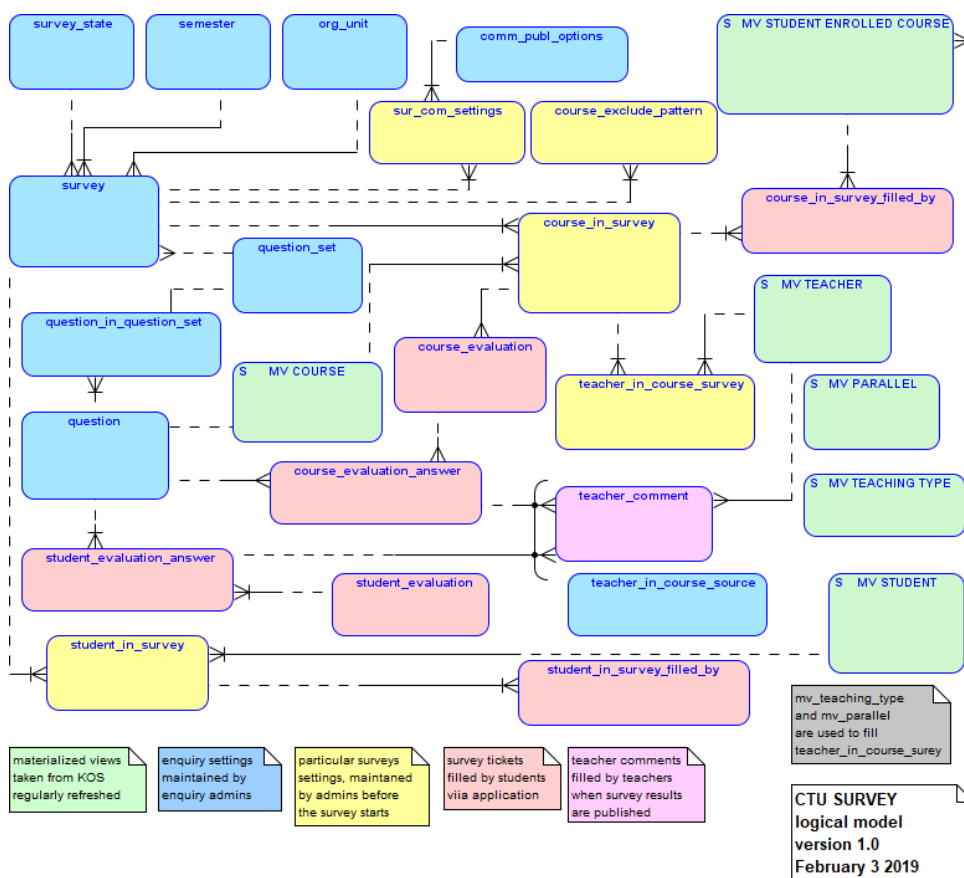
Databáze aplikace Anketa ČVUT využívá technologii Oracle. Následující podkapitola vysvětluje, jak jsou ankety v databázi reprezentovány.

3.1.1 Databázový model ankety

Databázový model ankety obsahuje potřebná data pro běh a vyhodnocení předmětových a nepředmětových anket na fakultách. Model ankety se celkově skládá z 20 tabulek obsažených ve schématu uživatele NOVA_ANKETA a 6 materializovaných pohledů ze schématu uživatele ANK_ANKETA. Tabulky lze dále podle jejich funkce rozdělit do 5 kategorií, které budou dále blíže popsány.

- Materializované pohledy z KOSu
- Nastavení ankety
- Nastavení účasti v anketě
- Odpovědi z anketních lístků
- Komentáře vyučujících

3. ADMINISTRÁTORSKÉ PROCESY V DATABÁZI



Obrázek 3.1: ER model databázového schématu ankety.

Materializované pohledy z KOSu

Pro každý běh ankety je na fakultě třeba udržovat aktuální přehled o studentech, předmětech a vyučujících na fakultě. Tato data jsou do databáze zkopírována ze systému KOS a jsou pravidelně aktualizována.

Entity *MV_STUDENT*, *MV_COURSE* a *MV_TEACHER* reprezentují studenty, předměty a vyučující. *MV_STUDENT_ENROLLED_COURSE* se váže k jednomu studentovi a jednomu předmětu a reprezentuje zápis předmětu studentem. *MV_TEACHING_TYPE* se váže k vyučujícímu a udává, zda vyučující vyučuje předmět jako přednášející, cvičící nebo zkoušející. *MV_PARALLEL* reprezentuje jednu paralelku v rámci semestru.

Nastavení ankety

Instance předmětové i nepředmětové ankety má definované určité vlastnosti, jako je například její stav, znění jejích otázek, na jaké fakultě probíhá atd. Nastavení provádí správce systému ve fázi přípravy ankety.

Entita *survey* reprezentuje instanci předmětové nebo nepředmětové ankety. Stav ankety, semestr, ve kterém anketa probíhá, a fakulta, na které anketa probíhá, jsou reprezentovány entitami *survey_state*, *semester* a *org_unit*.

Seznam otázek v anketě a jejich pořadí je reprezentováno entitami *question_set* a *question_in_question_set*, jejich přesné znění a to, zda je otázka hodnotová, nebo textová a zda se týká učitele, je pak definováno v entitě *question*.

Entita *comm_publ_options* reprezentuje identifikační údaje, které o sobě student může v anketním lístku sdílet.

Nastavení účasti v anketě

Ve fázi přípravy mají fakultní správci možnost přizpůsobit anketu pro potřeby své fakulty, jako například vyřadit z ní některé učitele nebo předměty.

Entity *course_in_survey* a *student_in_survey* představují předměty a studenty, kteří se účastní ankety. Entita *teacher_in_course_survey* reprezentuje vyučujícího, jenž se účastní ankety v rámci jednoho předmětu, který vyučuje. Účastníky lze z ankety vyloučit nastavením atributu entity.

Entita *course_exclude_pattern* reprezentuje regulární výraz, podle které lze vyřazovat předměty, pokud jejich kód danému výrazu odpovídá. Lze tak vyřadit celou skupinu předmětů, aniž by bylo nutné vyřazovat je individuálně.

Entita *sur_com_setting* reprezentuje identifikační údaje studenta, které se zobrazí v anketním lístku ankety a které student může zatrhnout.

Odpovědi z anketních lístků

Po otevření ankety mají studenti možnost vyplňovat anketní lístky. Odpovědi z anketních lístků, datum vyplnění a případné údaje, které se student rozhodl v anketě sdílet, se po odeslání ukládají do databáze.

Entita *course_evaluation* reprezentuje vyplněný anketní lístek předmětové ankety, odpovědi na otázky v anketním lístku jsou reprezentovány entitou *course_evaluation_answer*. Vyplněné anketní lístky nepředmětové ankety reprezentuje entita *student_evaluation* a odpovědi v anketním lístku entita *student_evaluation_answer*.

Entity *course_in_survey_filled_by* a *student_in_survey_filled_by* pak reprezentují datum, kdy student vyplnil anketní lístek předmětové a nepředmětové ankety.

Komentáře vyučujících

Vyučující mají po ukončení vyplňovací fáze a uzavření ankety možnost vyjádřit se k textovým odpovědím studentů.

Entita *teacher_comment* reprezentuje komentář vyučujícího k jedné odpovědi z anketního lístku předmětové nebo fakultní ankety.

3.2 Administrátorské činnosti v databázi

Pravidelné administrátorské činnosti probíhají v databázi aplikace pro aktualizaci dat a vytvoření nové ankety před jejím spuštěním a po jejím ukončení pro vyhodnocení výsledků.

3.2.1 Vyhodnocení ankety

Vyhodnocení ankety probíhá po ukončení vyplňovací fáze. V této fázi jsou zpracovávány odpovědi z anketních lístků do formátu vhodného k zveřejnění a zpracování zpětné vazby od studentů.

Vyhodnocení předmětové ankety

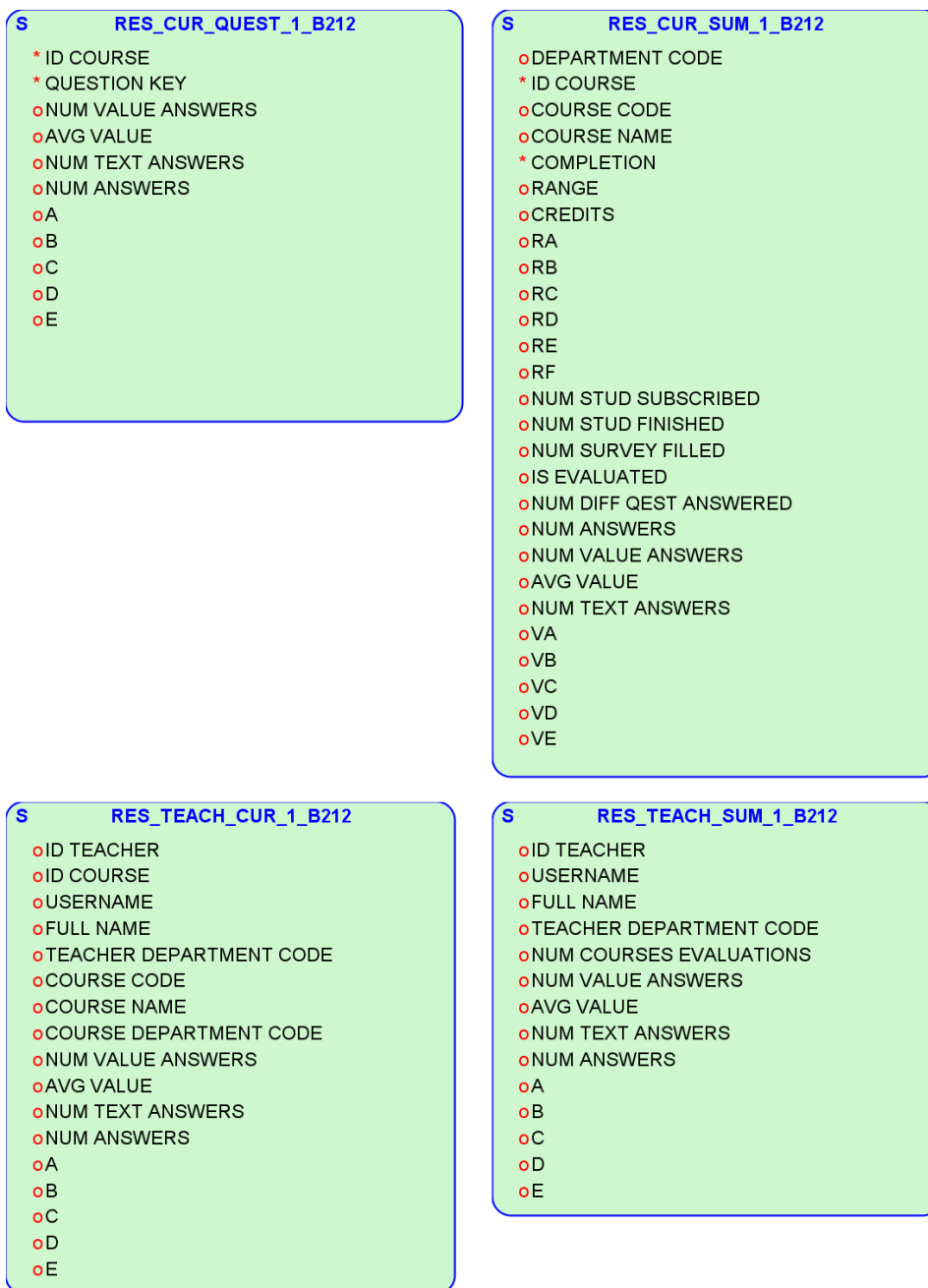
Pro vyhodnocení předmětové ankety jsou používány odpovědi z anketních lístků uložené v tabulce *course_evaluation_answer*.

Otázky předmětové ankety se mohou týkat buď předmětů, nebo vyučujících, což je možné rozlišit podle identifikačního klíče otázky.

Při vyhodnocování předmětových otázek jsou otázky v anketním lístku předmětu nejprve vyhodnoceny individuálně. Toto vyhodnocení je reprezentováno entitou *RES_CUR_QUEST*, která obsahuje údaj, o jaký předmět se jedná, klíč otázky, počet odpovědí na danou otázku a v případě hodnotové nebo kombinované otázky také distribuci známek a jejich průměrnou hodnotu.

Celkové hodnocení předmětu je reprezentováno entitou *RES_CUR_SUM*, které se vytváří sečtením a zprůměrováním odpovědí z *RES_CUR_QUEST*. Entita navíc obsahuje informace o předmětu, počet zapsaných studentů a počet studentů, kteří předmět dokončili, a jejich výsledky v předmětu.

Pro vyhodnocení vyučujících jsou vyučující nejprve vyhodnocováni v rámci jednoho předmětu, což reprezentuje entita *RES_TEACH_CUR*. Entita obsahuje informace o předmětu i vyučujícím, počet odpovědí, distribuci známek a jejich průměr. Celkové vyhodnocení pro vyučujícího napříč všemi předměty, které učil, je reprezentováno entitou *RES_TEACH_SUM*.

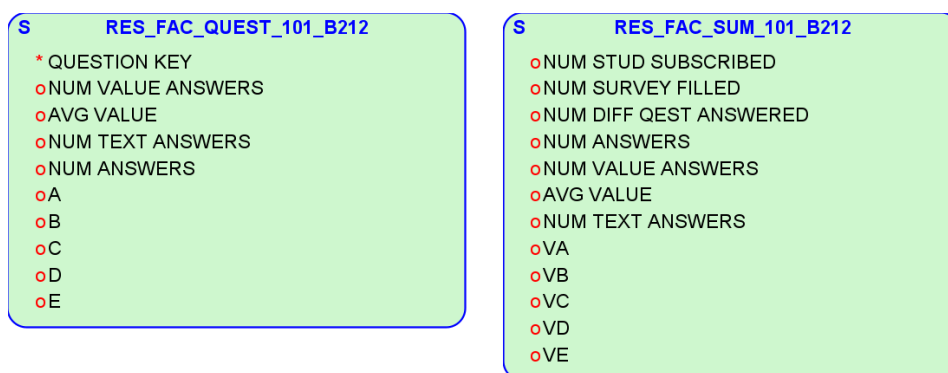


Obrázek 3.2: Tabulky pro vyhodnocení předmětové ankety.

Vyhodnocení nepředmětové ankety

Pro vyhodnocení nepředmětové ankety jsou používány odpovědi z anketních lístků uložené v tabulce *student_evaluation_answer*.

Vyhodnocení nepředmětové ankety probíhá obdobně jako vyhodnocení předmětů v předmětové anketě. Otázky týkající se fakulty jsou nejprve vyhodnoceny individuálně, což je reprezentováno entitou *RES_FAC_QUEST*. Celkové vyhodnocení fakulty pak reprezentuje entita *RES_FAC_SUM*.



Obrázek 3.3: Tabulky pro vyhodnocení nepředmětové ankety.

Archivace odpovědí z anketních lístků

Po vyhodnocení ankety jsou archivovány záznamy odpovědí z anketních lístků. Tabulky *RES_CUR_EVAL* a *RES_FAC_EVAL* obsahují kopii dat z tabulek *course_evaluation* a *student_evaluation*, což jsou záznamy o vyplnění anketního lístku předmětové nebo nepředmětové ankety a obsahují informace o studentovi a o tom, o jakou anketu ve kterém semestru se jedná. Samotné odpovědi z tabulek *course_evaluation_answer* a *student_evaluation_answer* jsou ukládány do *RES_CUR_ANSWER*, pokud se jedná o předmětové otázky z předmětové ankety, *RES_TEACH_ANSWER* pro učitelské otázky z předmětové ankety a *RES_FAC_ANSWER* pro otázky z nepředmětové ankety. Kromě údajů o studentovi a anketě a kopie odpovědí na otázky tabulky rovněž obsahují informace o tom, zda byla daná odpověď cenzurována. Informace o tom, kteří studenti se účastnili předmětové ankety a zda vyplnili anketní lístek předmětu, se ukládají do tabulky *RES_STUDENT_COURSE*.



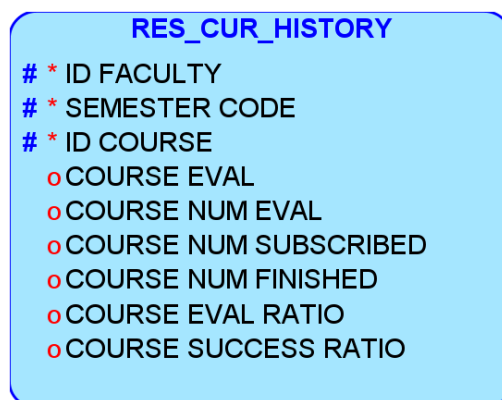
Obrázek 3.4: Tabulky pro archivaci odpovědí z anketních lístků.

3.2.2 Příprava ankety

Příprava ankety probíhá na konci semestru před jejím spuštěním.

Archivace vyplněnosti předchozí ankety

Před začátkem nové ankety jsou archivována data o vyplněnosti anketních lístků jednotlivých předmětů podle záznamů z *RES_CUR_SUM*. Pro tento účel slouží tabulka *RES_CUR_EVAL*, která zaznamenává hodnocení předmětu, počet ohodnocení, počet studentů, kteří si předmět zapsali a kteří předmět dokončili, a poměr průchodnosti semestru a vyplněnosti ankety.



Obrázek 3.5: Tabulky pro archivaci vyplněnosti předchozích anket.

Aktualizace materializovaných pohledů

Před začátkem ankety jsou aktualizovány materializované pohledy obsahující data o studentech, předmětech, zápisech předmětů, vyučujících a paralelkách podle aktuálních dat z KOSu.

Vytvoření nové ankety

V tabulce *survey* je vytvořena nová instance předmětové a nepředmětové ankety pro danou fakultu s aktuálním kódem semestru a pro obě ankety je vytvořen nový záznam v tabulce *sur_com_settings*. Pro předmětovou anketu jsou také definovány předměty vyloučené z ankety v tabulce *course_exclude_pattern*.

Dále dochází k přidání studentů jako účastníků ankety do tabulky *student_in_survey*. Student je do ankety na dané fakultě přidán, pokud na fakultě studuje nebo má zapsaný předmět, který je na ní vyučován.

3.2.3 Současné řešení

Administrativní činnosti v aplikaci Anketa ČVUT jsou v současné době částečně automatizovány pomocí anonymních PL/SQL skriptů, které vykonávají činnosti pro přípravu a vyhodnocení anket. Celkem se jedná o 11 skriptů.

Skripty pro přípravu ankety

copy-course-survey.sql Vytváří novou instanci předmětové ankety v tabulce *survey* podle záznamu předchozí ankety s novým kódem semestru a zadává nové údaje do tabulek *sur_com_settings* a *course_exclude_pattern* pro nový semestr.

copy-student-survey.sql Vytváří novou instanci nepředmětové ankety v tabulce *survey* podle záznamu předchozí ankety s novým kódem semestru.

load-student-into-student-survey.sql Přidá studenty, kteří studují na dané fakultě nebo na ní v daném semestru mají zapsaný předmět, do tabulky *student_in_survey* jako účastníky ankety na této fakultě.

rebuild-snapshots.sql Aktualizuje materializované pohledy v anketě podle aktuálních dat z KOSu.

rec-cur-history.sql Provádí archivaci vyplněnosti předchozí ankety do tabulky *RES_CUR_HISTORY*.

Skripty pro vyhodnocení ankety

create-eval.sql Provádí vyhodnocení předmětové ankety.

create-fac-eval.sql Provádí vyhodnocení nepředmětové ankety.

refresh-eval.sql Aktualizuje vyhodnocení předmětové ankety.

refresh-fac-eval.sql Aktualizuje vyhodnocení nepředmětové ankety.

dopln-programy.sql Doplnuje program studia studenta k odpovědím z anketních lístků, které jsou archivovány v tabulkách *RES_FAC_ANSWER*, *RES_CUR_ANSWER* a *RES_TEACH_ANSWER*.

check-created-view.sql Kontroluje počet záznamů vložených do tabulky *RES_STUDENT_COURSE* při vyhodnocování předmětové ankety.

3.2.4 Problémy se současným řešením

Současné řešení má několik nedostatků.

- Skripty je nutné spouštět jeden po druhém zvlášť.

3. ADMINISTRÁTORSKÉ PROCESY V DATABÁZI

- Ve skriptech je nutné pro každý semestr ručně upravit parametry.
- Pro jednu funkci existuje několik skriptů, které se liší pouze identifikátorem fakulty.
- Skripty je nutné spouštět přímo v databázi, veškerá zodpovědnost tedy spadá na správce systému.

Návrh řešení

Přestože je aplikace Anketa ČVUT v provozu již několik let, všechny její funkce ještě nejsou zcela automatizovány.

Jak bylo řečeno v předchozí kapitole, v současné době je pro provedení přípravy nebo vyhodnocení ankety potřeba provést sérii vícero úkonů. Je nutné ručně upravit PL/SQL skripty pro daný proces, připojit se do databáze aplikace a ručně skripty spustit, což může v současné době provést pouze správce systému Anketa ČVUT.

Pro zjednodušení provádění administrátorských činností bude správcovský modul aplikace rozšířen o nástroj, který umožňuje provádět přípravu a vyhodnocení ankety vzdáleně s možností specifikovat parametry, jako je kód fakulty a semestr. Díky tomu by správu ankety na fakultě mohl provádět přímo fakultní správce.

Vyhodnocení předmětové ankety

Fakulta
FIT

Ročník
2022/2023

Semestr
Letní

vyhodnotit

Obrázek 4.1: Náčrt uživatelského rozhraní pro vyhodnocení ankety.

Výstupem této práce bude integrace administrátorských procesů do systému

tak, aby je bylo možné spustit z backendové části aplikace. Tvorba nástroje ve správcovském modulu je předmětem budoucího vývoje, který není náplní této práce.

4.1 Požadavky

Rozšíření aplikace pro začlenění administrátorských procedur by mělo odpovídat následujícím funkčním a nefunkčním požadavkům.

Funkční požadavky

F1 – Aktualizace dat z KOSu Aplikace aktualizuje vybrané materializované pohledy podle aktuálních dat z KOSu.

- F1.1 – Aktualizace *MV_STUDENT*
- F1.2 – Aktualizace *MV_COURSE*
- F1.3 – Aktualizace *MV_TEACHER*
- F1.4 – Aktualizace *MV_STUDENT_ENROLLED_COURSE*
- F1.5 – Aktualizace *MV_TEACHING_TYPE*
- F1.6 – Aktualizace *MV_PARALLEL*

F2 – Vytvoření nové instance ankety Aplikace vytvoří novou anketu pro fakultu.

- F2.1 – Vytvoření nové instance předmětové ankety v tabulce *survey*
- F2.2 – Vytvoření nové instance nepředmětové ankety v tabulce *survey*
- F2.3 – Přidání studentů jakožto účastníků vytvořené ankety v tabulce *student_in_survey*
- F2.4 – Definice předmětů vyloučených z předmětové ankety v tabulce *course_exclude_pattern*
- F2.5 – Definice identifikačních údajů, které student může sdílet s odpověďmi v anketním lístku ankety, v tabulce *sur_com_settings*

F3 – Archivace vyplněnosti předchozích anket Výsledky anket z běhu minulého semestru budou archivovány do tabulky *res_cur_history*.

F4 – Vyhodnocení předmětové ankety Aplikace provede vyhodnocení předmětové ankety na fakultě *X* v semestru *Y*.

- F4.1 – Vytvoření a naplnění tabulky *RES_CUR_QUEST_X_Y*.
- F4.2 – Vytvoření a naplnění tabulky *RES_CUR_SUM_X_Y*.

- F4.3 – Vytvoření a naplnění tabulky *RES_CUR_EVAL_X_Y*.
- F4.4 – Vytvoření a naplnění tabulky *RES_CUR_ANSWER_X_Y*.
- F4.5 – Vytvoření a naplnění tabulky *RES_TEACH_CUR_X_Y*.
- F4.6 – Vytvoření a naplnění tabulky *RES_TEACH_SUM_X_Y*.
- F4.7 – Vytvoření a naplnění tabulky *RES_TEACH_ANSWER_X_Y*.
- F4.7 – Vytvoření a naplnění tabulky *RES_STUDENT_COURSE_X_Y*.

F5 – Vyhodnocení nepředmětové ankety Aplikace provede vyhodnocení nepředmětové ankety na fakultě *X* v semestru *Y*.

- F5.1 – Vytvoření a naplnění tabulky *RES_FAC_QUEST_X_Y*.
- F5.2 – Vytvoření a naplnění tabulky *RES_FAC_SUM_X_Y*.
- F5.3 – Vytvoření a naplnění tabulky *RES_FAC_EVAL_X_Y*.
- F5.4 – Vytvoření a naplnění tabulky *RES_FAC_ANSWER_X_Y*.

F6 – Aktualizace vyhodnocení předmětové ankety Aplikace provede aktualizaci výsledků předmětové ankety.

F7 – Aktualizace vyhodnocení nepředmětové ankety Aplikace provede aktualizaci výsledků nepředmětové ankety.

F8 – Logování Proběhlé činnosti budou zaznamenány včetně času spuštění a výsledku.

Nefunkční požadavky

N1 – Spouštění z backendu Administrátorské činnosti bude možné spouštět z backendové části aplikace.

4.2 Výběr technologií

Cílem této práce je začlenit procesy administrátorských činností do aplikace Anketa ČVUT.

V současnosti administrátorské procesy probíhají na úrovni databáze aplikace formou PL/SQL skriptů. Jako první z možných způsobů implementace se tak nabízí převedení skriptů do databázových procedur. Tento způsob řešení by vyžadoval pouze minimální úpravy zdrojového kódu, protože v současnosti používané skripty a databázové procedury používají stejný programovací jazyk – PL/SQL.

Protože databázové procedury neumožňují provádění DDL příkazů, bylo by v tomto případě nutné použít Dynamic SQL, a to buď formou nativního Dynamic SQL v podobě příkazu EXECUTE IMMEDIATE, nebo využitím funkcí balíčku DBMS_SQL.

```
create or replace PROCEDURE SELECT_TABLE( table_name IN varchar2 )
AS
BEGIN
    EXECUTE IMMEDIATE 'select * from ' || table_name;
END SELECT_TABLE;
```

Listing 3: Databázová procedura používající Dynamic SQL pro výkon DDL příkazu.

```
public void select_table( Connection con, String tableName ){
    CallableStatement statement = con.prepareCall(
        "{ call SELECT_TABLE( " + tableName + " )}"
    );
    statement.executeQuery();
}
```

Listing 4: Volání databázové procedury v jazyce Java pomocí třídy CallableStatement

Procedury by pak bylo možné volat z Java backendu databáze pomocí třídy CallableStatement z knihovny java.sql.

Alternativou k databázovým procedurám je implementace procesů podle PL/SQL skriptů přímo v backendové aplikaci. Jazyk Java, ve kterém je backendová aplikace napsána, umožňuje připojení k databázi pomocí JDBC API. JDBC API obsahuje balíček java.sql, pomocí kterého lze pracovat s daty v databázi prováděním SQL dotazů a spouštěním databázových procedur.[12] Jednou z funkcionalit tohoto balíčku, které by bylo možné použít, je třída PreparedStatement, která umožňuje zapsat SQL dotaz pro opakované použití a nahradit části dotazu proměnnými, které lze obměňovat.

```
public void select_table( Connection con, String tableName ){
    String query = "select * from ?";
    PreparedStatement statement = con.prepareStatement( query );
    statement.setString( 1, tableName );
    statement.executeQuery();
}
```

Listing 5: Provedení příkazu SQL v jazyce Java pomocí třídy PreparedStatement.

Po dohodě s vedoucím práce byla vybrána metoda implementace procedur v Oracle databázi. Důvodem pro toto rozhodnutí je především možnost použití již napsaných skriptů bez nutnosti přepisovat je do jiného programovacího jazyka.

Implementace

Následující kapitola popisuje implementaci databázových procedur v aplikaci Anketa ČVUT včetně vývojových nástrojů a specifických problémů, které bylo při vývoji nutné řešit.

5.1 Vývojové nástroje

5.1.1 Oracle SQL Developer

Pro implementaci databázových procedur byl použit nástroj SQL Developer distribuovaný společností Oracle. Pro tvorbu databázových schémat byla použita grafická nadstavba SQL Developeru Data Modeler.

5.1.2 Vývojové prostředí

Pro vývoj bylo použito databázové prostředí *dev* aplikace Anketa ČVUT. Toto prostředí obsahuje kopii dat z produkční databáze a je určeno k aktivnímu vývoji.

Výhodou použití vývojového prostředí je možnost pracovat při implementaci a testování se skutečnými daty, aniž by hrozilo jejich znehodnocení či ztráta nebo naopak ponechání neplatných testovacích dat v databázi. Kopie dat na vývojovém prostředí je navíc každý týden obnovována podle dat z produkční databáze.

5.2 Implementace procedur

Implementace spočívala v převedení původních anonymních PL/SQL skriptů do uložených procedur v Oracle databázi aplikace Anketa ČVUT. Pro každý skript byla vytvořena samostatná metoda. Pokud skript existoval ve více verzích pro jednotlivé fakulty, pak pro něj byla vytvořena jedna procedura, která přijímá kód fakulty jako parametr.

Celkem tak bylo vytvořeno dvanáct procedur – jedenáct z nich odpovídá původním skriptům plus procedura zprostředkávající funkci logování. Dále byly vytvořeny dva nové balíčky `new_sem` a `eval_sem`, do kterých byly procedury vloženy podle toho, zda jsou používány při přípravě anket pro aktualizaci dat a vytvoření anket, nebo po ukončení anket pro vyhodnocení a archivaci výsledků.

Tabulka 5.1: Struktura balíčků s novými procedurami.

package	procedura	původní skript
eval_sem	course_eval_create	create_eval
	course_eval_refresh	refresh_eval
	fac_eval_create	create_fac_eval
	fac_eval_refresh	refresh_fac_eval
	course_check_views	check_created_views
	fill_in_programs	doplň_programy
new_sem	course_eval_copy	copy_course_survey
	fac_eval_copy	copy_student_survey
	fac_student_load	load_students_into_student_survey
	course_add_history	rec_cur_history
—	snapshots_new_sem	rebuild_snapshots
	procedure_log	—

Přestože jsou procedury psány ve stejném jazyce jako původní skripty, a samotný kód tak zůstal prakticky beze změny, nebyla implementace procedur zcela triviální. Při implementaci bylo potřeba řešit zejména tři zásadní problémy:

- Vyřešit, jak v kódu provádět DDL příkazy, což je funkcionalita, kterou procedury neumožňují.
- Parametrizovat názvy tabulek a materializovaných pohledů, které jsou vytvářeny v rámci vyhodnocení semestru.
- Ošetřit přijímané parametry tak, aby procedury byly odolné vůči SQL Injection.

5.2.1 Implementace DDL příkazů

Jednou z hlavních překážek implementace PL/SQL skriptů do databázových procedur je fakt, že procedury neumožňují provádění DDL příkazů – to zahrnuje mimo jiné i vytváření, modifikaci a mazání tabulek a materializovaných pohledů ve schématu databáze. Pro řešení tohoto problému jsem použila dynamické SQL. Při použití statického SQL je nutné, aby byly SQL příkazy známe a validní v čase době kompilace programu. Dynamické SQL naproti tomu umožňuje konstruovat SQL příkazy za běhu programu.

```
execute immediate 'DROP MATERIALIZED VIEW
  ANK_ANKETA.MV_PARALLEL';
execute immediate 'DROP MATERIALIZED VIEW
  ANK_ANKETA.MV_COURSE';
execute immediate 'DROP MATERIALIZED VIEW
  ANK_ANKETA.MV_STUDENT';
execute immediate 'DROP MATERIALIZED VIEW
  ANK_ANKETA.MV_TEACHER';
execute immediate 'DROP MATERIALIZED VIEW
  ANK_ANKETA.MV_STUDENT_ENROLLED_COURSE';
execute immediate 'DROP MATERIALIZED VIEW
  ANK_ANKETA.MV_TEACHING_TYPE';
```

Listing 6: Použití příkazu EXECUTE IMMEDIATE pro smazání materializovaných pohledů.

5.2.2 Parametrizace názvů tabulek


Jedním z hlavních problémů při vyhodnocování ankety je potřeba zohlednit kód fakulty a semestru v názvu vytvářených tabulek. PL/SQL však neumožňuje používat proměnné v názvech tabulek, každý z vyhodnocovacích skriptů tak musel existovat v několika verzích, aby bylo možné vyhodnotit ankety na všech fakultách. Tyto skripty bylo navíc nutné pro každý semestr upravovat, aby byl kód semestru ve vytvářené tabulce aktuální.

Tento problém lze také vyřešit pomocí dynamic SQL. Příkaz EXECUTE IMMEDIATE přijímá řetězec znaků, který potom interpretuje jako SQL příkaz, je tedy možné do textového řetězce vkládat proměnné pomocí operace zřetězení.


```
begin
execute immediate 'drop materialized view '
  || course_question_view;
exception when others then null;
end;
```


Listing 7: Použití příkazu EXECUTE IMMEDIATE pro smazání pohledu, jehož jméno je známé až za běhu programu.


 fa-create-eval-8-b212.sql


 fbmi-create-eval-5-b212.sql


 fd-create-eval-4-b212.sql


 fel-create-eval-b212.sql

 fit-create-eval-1-b212.sql

 fjfi-create-eval-3-b212.sql

 fsi-create-eval-b212.sql

 fsv-create-eval-2-b212.sql

 muvs-create-eval-7-b212.sql

Obrázek 5.1: Skripty pro vyhodnocení předmětové ankety na jednotlivých fakultách pro letní semestr ročníku 2021/2022.

5.2.3 Ošetření parametrů

Přestože příkaz EXECUTE IMMEDIATE je velice silný nástroj a lze díky němu provádět příkazy, které statické SQL neumožňuje, může také učinit kód – a tím i celou aplikaci – zranitelnější vůči útokům. Protože je vstup uživatele vkládán přímo do kódu SQL příkazu, je možné, aby uživatel vložil do kódu svůj vlastní kód, který je poté spuštěn a vykonán. Uživatel by si tak mohl zobrazit data, která by měla být před běžným uživatelem aplikace utajena, nebo způsobit modifikaci či ztrátu dat v databázi.[4]

```

28
29     execute immediate
30         'create materialized view ' || student_course_view || '
31     as
32     select username, id_student, id_course,
33         course_completed course_completed,
34         survey_filled, course_excluded
35     from v_survey_course_student
36     where survey_key = ' || survey_id || '
37         and semester_code = '' || semester || ''
38     order by id_student, id_course';
39
40

```

Obrázek 5.2: SQL Injection varování v editoru Oracle SQL Developer v proceduře course_eval_create.

Přestože část aplikace, která bude spouštět administrátorské činnosti, bude využívána pouze interně, a riziko útoku by tedy mělo být minimální, je v rámci rozšiřitelnosti a znovupoužitelnosti kódu vhodné vstupy ošetřit.

Potenciální riziko představují argumenty, které mají formát textové hodnoty.

Mnoho procedur přijímá jako textový argument kód semestru. Tento argument je následně v SQL příkazu porovnáván s textovou hodnotou, pro ošetření tohoto vstupu tak je možné použít funkci ENQUOTE_LITERAL balíčku DBMS_ASSERT. Tato funkce upravuje vstupní hodnotu tak, aby byla vždy interpretována jako řetězec znaků.[13] Jakýkoli kód, který se uživatel pokusí do příkazu přidat, tak bude součástí této textové hodnoty a nikoliv vykonávaného kódu.





```
execute immediate 'create materialized view '  
    || student_course_view || '  
as  
select username, id_student, id_course,  
    course_comleted course_completed,  
    survey_filled, course_excluded  
from v_survey_course_student  
where survey_key = ' || survey_id || ' and semester_code = '  
    || DBMS_ASSERT.ENQUOTE_LITERAL( semester ) || '  
order by id_student, id_course';
```

Listing 8: Ošetření vstupu pomocí DBMS_ASSERT.ENQUOTE_LITERAL v proceduře course_eval_create.

Dalším textovým argumentem přijímaným procedurou je seznam klíčů předmětových a učitelských otázek v proceduře *course_eval_create*, který je používán pro rozeznání, zda je daná otázka z anketního lístku předmětovou nebo učitelskou otázkou. V tomto případě nelze pro ošetření použít funkci ENQUOTE_LITERAL, protože seznam klíčů není v SQL příkazu interpretován jako textový řetězec, nýbrž jako seznam číselných hodnot klauzule WHERE IN. Pro řešení tohoto problému bylo nutné nejprve řetězec znaků rozdělit na jednotlivé číselné hodnoty a následně je vložit do tabulky. Pro tento účel jsou při běhu procedury vytvářeny tabulky *course_quest* a *teacher_quest*. V kódu SQL příkazu je pak použit výběr dat z těchto tabulek.

```
BEGIN  
EXECUTE IMMEDIATE 'drop table course_quest';  
EXCEPTION when others then NULL;  
END;  
  
EXECUTE IMMEDIATE 'create table  
    course_quest( id number( 30 ),  
    constraint cq_pk primary key ( id ))';  
if l_course_questions is not null then  
FOR i IN ( SELECT to_number( column_value )  
as  
ID FROM xmltable( l_course_questions )) LOOP  
EXECUTE IMMEDIATE 'insert into course_quest  
    values( ' || i.ID || ' )';  
END LOOP;  
END IF;
```



Listing 9: Ošetření vstupu pomocí vložení hodnot do tabulky.

NOVA_ANKETA.COURSE_QUEST	NOVA_ANKETA.TEACHER_QUEST
P * ID NUMBER (30)	P * ID NUMBER (30)
 CQ_PK (ID)	 TQ_PK (ID)
 CQ_PK (ID)	 TQ_PK (ID)

Obrázek 5.3: Tabulky obsahující klíče předmětových a učitelských otázek pro potřeby procedury `course_eval_create`.

5.2.4 Logovací procedura

Jedním z požadavků na aplikaci byla funkce zaznamenání o spuštění procedury včetně času spuštění a stavu, v jakém procedura došla. Pro tento účel byla vytvořena procedura `procedure_log`. Tato procedura je volána ze všech ostatních procedur a jako parametr dostává jméno a stavový kód procedury, která ji zavolala. Tyto údaje společně s časovým údajem ukládá do tabulky `log_table`.

NOVA_ANKETA.LOG_TABLE	
P * DATETIME	TIMESTAMP
* EVENT	VARCHAR2 (100 BYTE)
* STATUS	VARCHAR2 (20 BYTE)
* CODE	VARCHAR2 (100 BYTE)
* USERNAME	VARCHAR2 (50 BYTE)
 LOG_PK (DATETIME)	
 LOG_PK (DATETIME)	

Obrázek 5.4: Tabulky pro záznamy proběhlých procedur.

5.3 Testování a nasazení

Vytvořené procedury byly testovány vedoucím práce před začátkem běhu ankety pro zimní semestr ročníku 2022/2023. V této době proběhlo také jejich nasazení do produkční databáze, kde byly použity k přípravě vytvářených anket.

Závěr

Cílem této práce bylo analyzovat a zdokumentovat administrátorské procesy v aplikaci Anketa ČVUT, navrhnout možnosti, jak administrátorské procesy spouštět z backendu aplikace, a jednu z možností zvolit, implementovat a zdokumentovat.

Výstupem této práce je rozšíření Oracle databáze, kterou aplikace používá, o spustitelné procedury, které zprostředkovávají procesy přípravy a vyhodnocení anket probíhajících v rámci aplikace Anketa ČVUT. Tyto procedury byly již použity pro přípravu a vyhodnocení ankety probíhající v zimním semestru ročníku 2022/2023. Lze tedy říci, že formulované cíle práce byly splněny. Správcovskou aplikaci tak bude v budoucnu možné rozšířit o nástroj, ze kterého bude možné administrátorské procesy spouštět.

Dalším přínosem této práce je také dokumentace některých dosud nepopsaných částí databáze aplikace Anketa ČVUT, které byly zdokumentovány v rámci analýzy administrátorských činností v databázi.

Bibliografie

1. POKORNÝ, Jaroslav; VALENTA, Michal. *Databázové systémy*. 3. vyd. Praha: Česká technika - nakladatelství ČVUT, 2020. ISBN 978-80-01-06696-6.
2. *PL/SQL Subprograms* [online]. Oracle, 2023 [cit. 2023-05-09]. Dostupné z: <https://docs.oracle.com/en/database/oracle/oracle-database/23/lnpls/plsql-subprograms.html>.
3. *Why can't we write ddl statement directly into the PL/SQL block* [online]. Stack Exchange, 2013 [cit. 2023-05-11]. Dostupné z: <https://dba.stackexchange.com/questions/37362/why-cant-we-write-ddl-statement-directly-into-the-pl-sql-block>.
4. *PL/SQL Dynamic SQL* [online]. Oracle, 2023 [cit. 2023-05-09]. Dostupné z: <https://docs.oracle.com/en/database/oracle/oracle-database/23/lnpls/dynamic-sql.html>.
5. *Hodnocení studia* [online]. ČVUT v Praze, Fakulta stavební, 2023 [cit. 2023-04-12]. Dostupné z: <https://portal.fsv.cvut.cz/anketa/>.
6. *Metodický pokyn č. 3/2022: Anketa ČVUT: dokumentace a metodika provozu systému* [online]. 10 2022. [cit. 2023-04-23]. Dostupné z: <https://www.cvut.cz/sites/default/files/content/d1dc93cd-5894-4521-b799-c7e715d3c59e/cs/20221007-metodicky-pokyn-c-32022.pdf>.
7. VALENTA, Michal. Otázky zařazené do ankety. *Anketa ČVUT* [online]. 04 2023 [cit. 2023-04-23]. Dostupné z: https://anketa.cvut.cz/reports/otazky/anketni_otazky_fit.html.
8. VALENTA, Michal. Anketní otázky - vysvětlení. *GitLab FIT* [online]. 01 2020 [cit. 2023-04-23]. Dostupné z: https://gitlab.fit.cvut.cz/anketa_cvut/nova-anketa-doc/-/wikis/Anketn%C3%AD-ot%C3%A1zky-vysv%C4%9Btlen%C3%AD.

9. ŠTĚCHA, Vojtěch. *Anketa ČVUT - verze 3.0 - vyplňování anketních lístků*. Praha, 2018. Diplomová práce. ČVUT v Praze, Fakulta informačních technologií, Katedra softwarového inženýrství.
10. *What is three-tier architecture?* [online]. International Business Machines Corporation [cit. 2023-04-10]. Dostupné z: <https://www.ibm.com/topics/three-tier-architecture>.
11. JUN, Jakub. *Anketa ČVUT verze 3 - modul pro správu anket - uživatelské rozhraní*. Praha, 2019. Bakalářská práce. ČVUT v Praze, Fakulta informačních technologií, Katedra softwarového inženýrství.
12. *Package java.sql* [online]. Oracle, 2023 [cit. 2023-05-10]. Dostupné z: <https://docs.oracle.com/javase/8/docs/api/java/sql/package-summary.html>.
13. *DBMS_ASSERT* [online]. Oracle, 2023 [cit. 2023-05-09]. Dostupné z: https://docs.oracle.com/en/database/oracle/oracle-database/19/arpls/DBMS_ASSERT.html.